経営情報研究
第17巻第2号(2009), 27－34ページ

**研究論文**

# 乗算法乱数の統計的検定

牧　野　　純

# Empirical Tests of Multiplicative Lagged Fibonacci Random Number Generators

Jun MAKINO

**【要　約】**乱数検定パッケージ TestU01 の BigCrush を用いて，さまざまな次数とビット長の乗算法乱数を検定した．次数が 3217 のように非常に大きいときには，33 ビットの乱数がすべての検定に合格する．しかし 607 次程度のよく利用される次数では，34 ビットが必要である．そして，今回調べたすべての次数について，48 ビットの乗算法乱数はすべての検定に合格した．これらの検定結果は，乗算法乱数を 32 ビット環境に実装するのに有用である．

## 1 はじめに

乗算法乱数は乗算型 Lagged Fibonacci 乱数とも呼ばれ，Marsaglia[3] によって提案された乱数発生法である．非常に簡潔な漸化式によって長周期で性質の良い擬似乱数を生成できることが知られている．ただし，その数列の最下位ビットは常に 1，すなわち奇数であり，つぎのビットは原始 3 項式による M 系列である．一般に，乗算型 Lagged Fibonacci 乱数は下位のビットが上位のビットよりランダムでない傾向がある．そこで，32 ビット全体がランダムであるような乱数が必要な場合，より長い語長の乗算法乱数を生成して，その上位ビットを用いることが好ましい．

乱数がランダムかどうかを調べるためには, 経験的検定を行う. 実際に乱数を発生してみて，さまざまな統計量を計算し，それらが真の乱数の分布と矛盾しないかどうかを統計的に検定するのである．このような目的で，多くの検定項目が考案されてきた．数列が真にランダムとみなせるためには，どのような見方をしても，それが真の乱数と同じ統計的性質を持たなければならないからである．

今日では, こうした検定項目をいくつかセットにした乱数検定パッケージが提供されている．こうしたパッケージは，パッケージ内のすべての項目の検定に合格する乱数は，パッケージに含まれない他の多くの検定項目についても合格することを目標に設計されている．

このような検定パッケージのうち，最近よく使われるのが L'Ecuyer らによる TestU01 である [2]．TestU01 には SmallCrush, Crush, BigCrush と呼ばれる 3 種類のセットが用意されていて，この順に簡便な検定から本格的な検定までが可能となっている．中でも BigCrush は $2^{38}$ 個の乱数を用い, 106 の検定項目について 160 個の統計量の $p$ 値を計算するという徹底した検定セットで，そのすべての検定に合格する乱数発生法はごく限られている．BigCrush に合格することは新しい乱数発生法を開発するときの目標となっている．

L'Ecuyer らは，TestU01 の解説論文 [2] の中で，多くの乱数発生法を TestU01 により検定してその結果を公表している．その中で，64 ビットの乗算型 Lagged Fibonacci 乱数が BigCrush を含むすべての検定に合格することを報告している．32 ビットの乗算法については報告がないが，それが奇数の乱数であることから，すべての検定に合格することは考えられない．では，64 ビットと 32 ビットとの中間のビット数の乗算法乱数の場合はどうであろうか．それを調べるのがこの論文の目的である．

64 ビットの演算は 32 ビット環境では発生に少々時間がかかる．もし，より少ないビット数の乱数でも性質の良いことがわかれば，32 ビット環境で効率的に利用でき，しかも下位ビットまで性質の良い乗算法乱数を開発できる可能性がある．本研究の結果はこのような開発に基礎的な動機を与えるものと考えられる．

## 2　乗算法乱数

乗算法乱数は正確には乗算型 Lagged Fibonacci 乱数と呼ばれ，Marsaglia[3] によって提案された乱数発生法である．$p$ 個の初期値 $x_0, x_1,\ldots,x_{p-1}$ と $p$ 次の漸化式

$$x_i = x_{i-p} * x_{i-q} \bmod 2^e$$

によって定まる $e$ ビット整数の数列$\langle x_i\rangle$を乱数として利用する．ここで $p, q$ は $p>q$ を満たす整数で，ラグと呼ばれる．$x^p + x^q + 1$ は $GF(2)$上の原始 3 項式でなければならない．また，$p$ 個の初期値の最下位ビットはすべて 1（つまり初期値はすべて奇数）であり，つぎのビットはすべてが 0 ではいけない（少なくとも 1 つは 1 である）．これらの条件を満たすとき，乗算法乱数の周期は $(2^p-1)2^{e-3}$ となる．

乗算法乱数は非常に簡単な漸化式によって発生されるのにもかかわらず，下位ビットを除いて性質が良いことが知られている．また，数列の途中をスキップして遠隔項を効率良く計算できるアルゴリズムも開発されている [4]．この数列の任意の部分にジャンプできるという性質は，とくに並列計算機でのモンテカルロシミュレーションにとって好都合である．

$e$ はコンピュータの整数のビット長にとるのが普通である．たとえば 32 ビットのコンピュータでは $e = 32$，64 ビットのコンピュータなら $e = 64$ である．ただし，64 ビットのコンピュータでも $e = 32$ とする場合も多い．その最大の理由は，プログラムの互換性の問題であろう．シミュレーションのプログラムは，どのような処理系であろうと，同じ結果が得られることが好ましいのである．

## 3　検定の方法と結果

TestU01 には汎用検定セットとして SmallCrush, Crush, BigCrush の 3 種類が用意されている．高性能の PC を使った場合の所要時間はそれぞれ，数秒，数十分，数時間である．

通常の乱数の検定には，まず SmallCrush を適用し，それに合格すれば Crush を実行し，それも合格であれば最後に BigCrush にかけるという使い方が行われる．これは検定の効率を考えてのことである．ほとんどの乱数発生法は SmallCrush や Crush で何らかの規則性を露呈してしまい，時間のかかる BigCrush で検定する必要がないからである．

しかし，BigCrush に合格する乱数は SmallCrush や Crush には難なく合格するのが普通である．そこで，今回はすべての乱数の検定をいきなり BigCrush からすることにした．パラメータを少しずつ変えて乱数を調べていくとき，乱数の性質も少しずつ変化する．そのため，完全に不合格になる（おそらく SmallCrush または Crush でも不合格になるであろう）場合の BigCrush の検定結果も，合格・不合格があいまいな場合の検定のために有益だからである．この点に関しては，この節で再検定の議論をするとき，詳しく解説する．

### 3.1 BigCrush検定のまとめ

いろいろなサイズのラグ $(p, q)$ について, ビット数を 32 ビットから 48 ビットまで変えながら, BigCrush による検定を行った.

BigCrush には検定の対象となる 32 ビット乱数を入力する．そこで乱数は，内部的には 64 ビットで生成し，検定に必要な 32 ビットを返すようにした．たとえば 40 ビット乱数の検定では，内部で生成した 64 ビットのうち上位 24 ビットを捨てて 40 ビット乱数とし，さらにその下位 8 ビットを捨てて上位 32 ビットを返す.

検定の結果として出力される $p$ 値は，L'Ecuyer らの論文 [2] にしたがって，つぎの 3 種類の判定のいずれかに分類する.

合格： $[10^{-4}, 1 - 10^{-4}]$
疑わしい： $[10^{-10}, 10^{-4}) \cup (1 - 10^{-4}, 1 - 10^{-10}]$
不合格： $[0, 10^{-10}) \cup (1 - 10^{-10}, 1]$

判定の基準となる $10^{-4}$ や $10^{-10}$ の値の選び方はもちろん確定したものではない．通常の統計的検定の常識からすると，これらの有意水準はあまりに小さすぎるように思われるが，1 回の検定で 160 個の $p$ 値を求める BigCrush，それを数百回繰り返す今回のような大規模な検定では水準をこのくらいの値にしておかないと，偶然「不合格」または「疑わしい」ような $p$ 値を出力する回数が多くなりすぎて，結果の分析が不可能になる．また，乱数の検定の場合，その数学的な構造が検定によって検出されることになるので，不合格の場合には決定的に異常な $p$ 値として観測されることが多いので，優位水準をこの程度に小さくしても検出力が弱くなることはあまり考えられない.

表 1 に検定の結果を示す．表側の値はラグ $(p, q)$ である．上から下へ $p$ が小さい順に並べてあるが，$p = 127$ の場合だけはすべての原始 3 項式（5 通り）について検定を行った．これは乱数の性質が $q$ の値にどう依存するかを調べるためである.

これらの漸化式は L'Ecuyer ら [2] が 64 ビットの場合について検定した 4 種類の乱数$(p, q) =$ (17,5), (55,24), (607,273), (1279,861)に Knuth の教科書 [1] にある乱数と，あとは原始 3 項式の一覧表 [5, 6] から適当に補充したものである．$(p, q)$と$(p, p-q)$の漸化式は等価であるが，今回検定したものには $q < p/2$ のものと $q > p/2$ のものが混ざっている．これは，漸化式を上記の 3 種類の文献から選んだことによる.

表の中の記号は各乱数についての BigCrush の検定結果を表すもので，つぎのような意味をもつ.

F： 少なくとも 1 つの検定項目が不合格
S： （不合格ではないが）少なくとも 1 つの検定項目が疑わしい
．： すべての検定項目が合格

この表から，漸化式の次数 $p$ が大きいほど短いビット長で検定に合格することがわかる．一

方，$p = 127$ の 5 種類の乱数からわかるように，BigCrush の検定結果はもう一つのラグ $q$ にはそれほど依存しないようである．しかし，これ以上の解釈をする前に，この検定であいまいな結果に終わったいくつかの点について，再検定をする必要がある．

表 1：BigCrush 検定のまとめ

| p | q | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 17 | 5 | F | F | F | F | F | F | F | F | F | F | F | F | F | S | . | . | . |
| 55 | 24 | F | F | F | F | F | F | F | F | . | . | . | . | . | . | . | . | . |
| 89 | 38 | F | F | F | F | F | F | F | . | . | . | . | . | . | . | . | . | . |
| 100 | 37 | F | F | F | F | F | F | F | . | . | . | . | . | . | . | . | . | . |
| 127 | 1 | F | F | F | F | F | F | . | S | . | . | . | . | . | . | . | . | . |
| 127 | 7 | F | F | F | F | F | F | . | . | . | . | . | . | . | . | . | . | . |
| 127 | 15 | F | F | F | F | F | F | . | . | S | . | . | . | . | . | . | . | . |
| 127 | 30 | F | F | F | F | F | F | . | . | . | . | . | . | . | . | . | . | . |
| 127 | 63 | F | F | F | F | F | F | . | . | S | . | . | . | . | . | . | . | . |
| 145 | 52 | F | F | F | F | F | S | . | . | . | . | . | . | . | . | . | . | . |
| 201 | 79 | F | F | F | F | S | . | . | . | . | . | . | . | . | . | . | . | . |
| 258 | 83 | F | F | S | . | . | . | . | . | . | . | . | . | . | . | . | S | . |
| 274 | 67 | F | F | S | . | . | . | . | . | . | S | . | . | . | . | . | . | . |
| 300 | 91 | F | F | S | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 337 | 147 | F | F | S | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 378 | 107 | F | F | . | . | S | . | . | . | . | . | . | . | . | . | . | . | . |
| 607 | 273 | F | F | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 1279 | 861 | F | S | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 2281 | 1029 | F | S | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 3217 | 576 | F | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 9689 | 4187 | F | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 19937 | 7083 | F | . | . | . | . | . | . | . | . | . | . | . | . | . | S | . | . |
| 23209 | 9739 | F | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |

### 3.2 再検定と検定結果の解釈について

「疑わしい」と判定された検定項目については，乱数の異なるサンプルを用いて検定を何回か繰り返して，それが偶然によるものか，あるいは乱数の規則性を示すものなのかを判定する必要がある．

$(p, q)$ を固定してビット数を増やしていくと徐々に合格する検定項目が多くなっていく．ある検定項目について，ビット数が小さいとき不合格だった $p$ 値が，ビット数の増加とともに通常の値をとるようになる．

たとえば，$(p, q) = (15,7)$ の場合，44 ビット以下で「不合格」だったものが，45 ビットでは

「疑わしい」になり，46 ビットでは「合格」の判定になっている．そこで，45 ビットの場合について，「疑わしい」結果を出した検定項目について，検定を 5 回繰り返したところ，$1.7\times10^{-9}$, $1.9\times10^{-8}$, $3.4\times10^{-11}$, $2.5\times10^{-8}$, $4.2\times10^{-9}$ という $p$ 値が得られ，最初の「疑わしい」の判定は「不合格」と解釈すべきであることがわかった．このような境界のビット数の場合にあらわれる「疑わしい」は，多くの場合，乱数にまだ規則性が残っていることの結果である．

さらに「合格」についても，より詳しく調べると実は $p$ 値に異常が見られることもある．たとえば，$(p,q)$ = (55,24) の場合，39 ビット以下の場合「不合格」だったものが 40 ビットのときは「合格」になっている．しかし，この乱数について 35 番目の検定項目（Gap 検定の一つ）を 5 回繰返したところ, 0.04, 0.04, $7.5\times10^{-3}$, 0.15, $2.5\times10^{-3}$ という $p$ 値を得た．これらは個々に見ればいずれも「合格」と判定されるが，全体として見ると明らかに小さい値に偏っている．実際，100 回繰り返してみたところ，5 回で「疑わしい」という結果が得られた．確率 $10^{-4}$ でしか起こらないことが 100 回中 5 回も起こっているのである．したがって，$(p,q)$ = (55,24) の 40 ビットの場合も「不合格」と解釈すべきであろう．

合否の境界についてのこのような検討と，表 1 で「疑わしい」の判定結果のものについての再検定をすませた後の結果を，表 2 に示す．ここでは，記号「F」は BigCrush で少なくとも一つ以上の検定項目で「不合格」となったことを示し（表 1 と同じ），記号「f」は，BigCrush では不合格でなかったものの，「疑わしい」と判定された検定項目，あるいは他の乱数の検定結果から再検討が必要と思われる検定項目について，再検定を繰り返した結果により，「合格」とはみなせないことを示している．一方，記号「.」は一応「合格」であるが，その解釈には注意が必要である．BigCrush により再検定の必要な項目がすべて判明するわけではない．また再検定も，必要な項目についての検定を 5 回繰り返して結果を目視で判断するといった簡易的な検定で済ませている．このため，「.」は「今回の検定では不合格とする根拠が見つからなかった」という意味と解釈しなければならない．

## 4 おわりに

32 ビットよりビット長を長くして，乗算法乱数を TestU01 の BigCrush で検定した．漸化式の次数が $p$ = 3217 のように非常に大きいときには，33 ビットの乱数がすべての検定に合格する．しかし $p$ = 607 程度のよく利用される次数では，35 ビットが必要である．そして，今回調べたすべての $p$ にわたって 48 ビットの乱数はすべての検定に合格する．

これらのことは 32 ビット環境に乗算法乱数をインプリメントする際に，心得ておくべきことである．乱数の性質に関する限り，語長を 64 ビットまで伸張する必要はないのである．

この論文では再検定は 5 回繰り返して，目視で異常がないかを確認するにとどめた．もちろん，より徹底的な再検定も可能である．たとえば，再検定を 100 回繰り返し，出力された 100 個の $p$ 値を Kolmogorov-Smirnov 検定により一様分布と比較すればよい．しかし，このような再検定を今回の検定結果全体に適用するのは労多くして益は少ない．もし，特定の漸化式で正

表 2: 再検定の結果を考慮した結果

| p | q | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 17 | 5 | F | F | F | F | F | F | F | F | F | F | F | F | F | f | f | . | . |
| 55 | 24 | F | F | F | F | F | F | F | F | f | . | . | . | . | . | . | . | . |
| 89 | 38 | F | F | F | F | F | F | F | f | . | . | . | . | . | . | . | . | . |
| 100 | 37 | F | F | F | F | F | F | F | f | . | . | . | . | . | . | . | . | . |
| 127 | 1 | F | F | F | F | F | F | f | . | . | . | . | . | . | . | . | . | . |
| 127 | 7 | F | F | F | F | F | F | f | . | . | . | . | . | . | . | . | . | . |
| 127 | 15 | F | F | F | F | F | F | f | . | . | . | . | . | . | . | . | . | . |
| 127 | 30 | F | F | F | F | F | F | f | . | . | . | . | . | . | . | . | . | . |
| 127 | 63 | F | F | F | F | F | F | f | . | . | . | . | . | . | . | . | . | . |
| 145 | 52 | F | F | F | F | F | f | f | . | . | . | . | . | . | . | . | . | . |
| 201 | 79 | F | F | F | F | f | . | . | . | . | . | . | . | . | . | . | . | . |
| 258 | 83 | F | F | f | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 274 | 67 | F | F | f | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 300 | 91 | F | F | f | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 337 | 147 | F | F | f | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 378 | 107 | F | F | f | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 607 | 273 | F | F | f | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 1279 | 861 | F | f | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 2281 | 1029 | F | f | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 3217 | 576 | F | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 9689 | 4187 | F | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 19937 | 7083 | F | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 23209 | 9739 | F | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |

確な検定を行う必要があれば，そのとき，その漸化式だけに対して，こうした厳密な検定を行えば十分であろう．いまは，表 2 に示された結果を見て，本当に良い性質の乱数を作るためにはさらに 1 ビット程度が必要かもしれないと，慎重に解釈するのが妥当であろう．

今回の検定は，3.0GHz の Intel Core i7 プロセッサ搭載の 64 ビット linux PC で，gcc コンパイラを用いて行った．この環境で BigCrush を 1 つ実行する時間は，3 時間 40 分前後であった．ただし，4 コアのプロセッサなので，4 つの検定を並列に実行することができた．

## 参考文献

[1] Knuth, D. E. 1998. *The Art of Computer Programming*. Vol. 2: Seminumerical Algorithms, 3rd ed. Addison-Wesley, Reading, MA.

[2] L'Ecuyer, P. and Simard, R. 2007. TesuU01: A C Libray for Empirical Testing of RNGs, *ACM Transactions on Mathematical Software*. 33, Article 22.（ソフトウェアは http://www.iro.umontreal.ca/~simardr/testu01/tu01.html よりダウンロードできる.）

[3] Marsaglia, G. 1985. A current view of random number generators. In *Computer Science and Statistics, Sixteenth Symposium on the Interface*. Elsevier Science Publishers, North-Holland, Amsterdam, The Netherlands. 3-10.

[4] Makino, J. 1995. The Equipartition Method for Parallel Generation of Random Numbers. Ph.D. thesis, Osaka University.

[5] Zierler, N and Brillhart, J. 1968. On primitive trinomials (Mod 2). *Information and Control*. 13. 541-554.

[6] Zierler, N and Brillhart, J. 1969. On primitive trinomials (Mod 2), Ⅱ. *Information and Control*. 14. 566-569.